APOSTILA PRÁTICA DE LINUX

Comandos essenciais para Técnicos de Informática, Redes e Infraestrutura

---

Apresentação

O Linux está presente em servidores web, servidores de arquivos, equipamentos de rede, sistemas de monitoramento, ambientes de virtualização, containers, serviços em nuvem e dispositivos de infraestrutura.

Para um técnico de informática, conhecer Linux não significa apenas saber instalar o sistema operacional.

É necessário saber:

- navegar pelo sistema de arquivos;
- criar e remover arquivos e diretórios;
- localizar informações;
- administrar usuários e grupos;
- controlar permissões;
- instalar softwares;
- controlar serviços;
- verificar processos;
- diagnosticar problemas de rede;
- testar conectividade;
- verificar portas;
- analisar armazenamento;
- consultar logs;
- acessar servidores remotamente;
- configurar serviços;
- monitorar recursos;
- automatizar tarefas.

Esta apostila apresenta os principais comandos utilizados no dia a dia de um técnico de Linux, redes e infraestrutura.

Como estudar esta apostila

Para cada comando, procure entender quatro coisas:

1. O que o comando faz?

2. Qual é sua sintaxe?

3. Em que situação um técnico utilizaria esse comando?

4. O que acontece quando o comando é executado?

Sempre que possível, execute os exemplos em um ambiente de laboratório.

> **Atenção:** alguns comandos alteram configurações, excluem arquivos ou interrompem serviços. Nunca execute comandos destrutivos em um servidor de produção sem compreender sua finalidade.

UNIDADE 1 — O TERMINAL LINUX

1.1 O que é o terminal?

O terminal é uma interface que permite interagir diretamente com o sistema operacional por meio de comandos.

Um exemplo:

aluno@servidor:~$

Nesse exemplo:

- `aluno` → usuário conectado;
- `servidor` → nome da máquina;
- `~` → diretório pessoal do usuário;
- `$` → usuário comum.

Quando o usuário é `root`, normalmente aparece:

root@servidor:~#

O símbolo # indica que o usuário possui privilégios administrativos.

UNIDADE 2 — NAVEGAÇÃO PELO SISTEMA

2.1 pwd

Função

Mostra o diretório atual.

Comando

pwd

Exemplo

$ pwd
/home/aluno

Aplicação

É muito útil para saber em qual diretório você está antes de executar comandos.

2.2 ls

Função

Lista arquivos e diretórios.

Comando

ls

Exemplos

Listar arquivos:

ls

Mostrar detalhes:

ls -l

Mostrar arquivos ocultos:

ls -a

Mostrar arquivos ocultos e detalhes:

ls -la

Mostrar tamanhos de forma legível:

ls -lh

Exemplo prático

ls -lah /var/log

2.3 cd

Função

Muda o diretório atual.

Exemplos

cd /etc

Ir para o diretório pessoal:

cd ~

Voltar um diretório:

cd ..

Voltar ao diretório anterior:

cd -

2.4 tree

Função

Exibe a estrutura de diretórios em forma de árvore.

Exemplo

tree /etc

Pode ser necessário instalar:

apt install tree

UNIDADE 3 — CRIAÇÃO E MANIPULAÇÃO DE ARQUIVOS

3.1 mkdir

Função

Cria diretórios.

Exemplo

mkdir documentos

Criar vários diretórios:

mkdir documentos imagens backups

Criar uma estrutura completa:

mkdir -p projeto/site/css

3.2 touch

Função

Cria um arquivo vazio ou atualiza sua data de modificação.

Exemplo

touch teste.txt

Criar vários arquivos:

touch arquivo1.txt arquivo2.txt arquivo3.txt

3.3 cp

Função

Copia arquivos e diretórios.

Copiar arquivo

cp arquivo.txt backup.txt

Copiar para outro diretório:

cp arquivo.txt /home/aluno/

Copiar diretório:

cp -r projeto /backup/

3.4 mv

Função

Move ou renomeia arquivos.

Renomear:

mv antigo.txt novo.txt

Mover:

mv arquivo.txt /tmp/

3.5 rm

Função

Remove arquivos.

rm arquivo.txt

Remover vários arquivos:

rm arquivo1.txt arquivo2.txt

Remover diretório vazio:

rmdir pasta

Remover diretório e conteúdo:

rm -r pasta

⚠️ Atenção

O comando:

rm -rf

pode apagar grandes quantidades de dados sem solicitar confirmação.

Nunca execute comandos destrutivos sem verificar cuidadosamente o caminho.

UNIDADE 4 — VISUALIZAÇÃO E MANIPULAÇÃO DE TEXTOS

4.1 cat

Função

Exibe o conteúdo de arquivos.

cat arquivo.txt

Exemplo:

cat /etc/hostname

4.2 less

Função

Permite visualizar arquivos grandes página por página.

less /var/log/syslog

Teclas úteis:

- `↑` → sobe;
- `↓` → desce;
- `PageUp` → página anterior;
- `PageDown` → próxima página;
- `/texto` → pesquisa;
- `q` → sair.

4.3 head

Função

Mostra o início de um arquivo.

head arquivo.txt

Mostrar as primeiras 20 linhas:

head -n 20 arquivo.txt

4.4 tail

Função

Mostra o final de um arquivo.

tail arquivo.txt

Muito utilizado para análise de logs.

tail /var/log/syslog

Acompanhar um log em tempo real:

tail -f /var/log/syslog

Esse comando é extremamente útil para técnicos de infraestrutura.

UNIDADE 5 — LOCALIZAÇÃO DE ARQUIVOS

5.1 find

Função

Localiza arquivos e diretórios.

Procurar um arquivo:

find /home -name "teste.txt"

Procurar arquivos `.log`:

find /var -name "*.log"

Procurar arquivos maiores que 100 MB:

find / -type f -size +100M

Procurar arquivos modificados nas últimas 24 horas:

find /var -type f -mtime -1

5.2 which

Função

Mostra onde está localizado um executável.

which python3

Exemplo:

/usr/bin/python3

5.3 whereis

Função

Localiza arquivos relacionados a um programa.

whereis ssh

UNIDADE 6 — PERMISSÕES

Um dos conhecimentos mais importantes para um administrador Linux é entender permissões.

Exemplo:

-rwxr-xr--

As permissões são divididas em:

usuário | grupo | outros

Representação:

r = leitura
w = escrita
x = execução

Valores numéricos:

r = 4
w = 2
x = 1

6.1 chmod

Função

Altera permissões.

Exemplo:

chmod 755 script.sh

Significa:

Usuário: rwx = 7
Grupo: r-x = 5
Outros: r-x = 5

Dar permissão de execução:

chmod +x script.sh

Remover escrita de outros usuários:

chmod o-w arquivo.txt

6.2 chown

Função

Altera proprietário.

chown aluno arquivo.txt

Alterar usuário e grupo:

chown aluno:alunos arquivo.txt

Alterar recursivamente:

chown -R aluno:alunos /home/aluno/projeto

6.3 chgrp

Função

Altera o grupo proprietário.

chgrp alunos arquivo.txt

UNIDADE 7 — USUÁRIOS E GRUPOS

7.1 whoami

Função

Mostra o usuário atual.

whoami

7.2 id

Função

Mostra informações do usuário.

id

Exemplo:

id aluno

7.3 who

Função

Mostra usuários conectados.

who

7.4 w

Função

Mostra usuários conectados e informações sobre suas sessões.

w

7.5 adduser

Função

Cria um usuário.

adduser aluno

O Debian normalmente apresenta perguntas para configurar o usuário.

7.6 userdel

Função

Remove um usuário.

userdel aluno

Remover usuário e seu diretório pessoal:

userdel -r aluno

7.7 passwd

Função

Altera a senha.

passwd aluno

7.8 groupadd

Função

Cria um grupo.

groupadd turma_informatica

7.9 usermod

Função

Modifica informações de um usuário.

Adicionar usuário a um grupo:

usermod -aG turma_informatica aluno

Verificar grupos:

groups aluno

UNIDADE 8 — PROCESSOS

Um processo é um programa em execução.

8.1 ps

Função

Lista processos.

ps

Mostrar todos os processos:

ps aux

Procurar um processo:

ps aux | grep apache2

8.2 top

Função

Mostra processos e utilização de recursos em tempo real.

top

Permite observar:

- CPU;

- memória;
- processos;
- carga do sistema.

8.3 htop

Função

Uma interface mais amigável para monitoramento de processos.

Instalação:

apt install htop

Execução:

htop

8.4 kill

Função

Envia um sinal para um processo.

Primeiro descubra o PID:

ps aux | grep programa

Depois:

kill 1234

Em situações específicas:

kill -9 1234

⚠️ Atenção

`kill -9` força o encerramento do processo e deve ser usado somente quando necessário.

UNIDADE 9 — SERVIÇOS E SYSTEMD

Em servidores Debian, o `systemd` é utilizado para administrar muitos serviços.

9.1 systemctl status

Verificar o estado de um serviço:

systemctl status ssh

Exemplo:

systemctl status nginx

9.2 systemctl start

Iniciar:

systemctl start nginx

9.3 systemctl stop

Parar:

systemctl stop nginx

9.4 systemctl restart

Reiniciar:

systemctl restart nginx

9.5 systemctl reload

Recarregar configuração sem necessariamente reiniciar o processo:

systemctl reload nginx

9.6 systemctl enable

Fazer um serviço iniciar automaticamente:

systemctl enable nginx

9.7 systemctl disable

Desativar inicialização automática:

systemctl disable nginx

9.8 Verificar se um serviço está ativo

systemctl is-active nginx

UNIDADE 10 — LOGS DO SISTEMA

Logs são fundamentais para diagnóstico.

10.1 journalctl

Função

Consulta os registros do `systemd`.

journalctl

Mostrar logs recentes:

journalctl -n 50

Acompanhar em tempo real:

journalctl -f

Ver logs de um serviço:

journalctl -u nginx

Ver somente mensagens desde o boot atual:

journalctl -b

10.2 /var/log

Grande parte dos registros tradicionais fica em:

/var/log

Listar:

ls -lah /var/log

Exemplo:

cat /var/log/auth.log

UNIDADE 11 — GERENCIAMENTO DE PACOTES

No Debian, o principal sistema de gerenciamento de pacotes utiliza o APT.

11.1 apt update

Atualiza a lista de pacotes disponíveis.

apt update

11.2 apt upgrade

Atualiza os pacotes instalados.

apt upgrade

11.3 apt install

Instala um programa.

apt install nginx

Exemplo:

apt install htop

11.4 apt remove

Remove um programa:

apt remove nginx

11.5 apt purge

Remove o programa e seus arquivos de configuração gerenciados pelo pacote:

apt purge nginx

11.6 apt search

Pesquisa pacotes:

apt search nginx

11.7 apt show

Mostra informações de um pacote:

apt show nginx

11.8 apt autoremove

Remove dependências que não são mais necessárias:

apt autoremove

UNIDADE 12 — REDE

Esta é uma das unidades mais importantes para um técnico de infraestrutura.

12.1 ip addr

Função

Mostra interfaces e endereços IP.

ip addr

Forma resumida:

ip a

12.2 ip link

Mostra as interfaces de rede:

ip link

12.3 ip route

Mostra a tabela de roteamento:

ip route

Exemplo:

default via 192.168.1.1 dev eth0

Isso indica a rota padrão.

12.4 ping

Função

Testa conectividade.

ping 8.8.8.8

Testar um servidor:

ping 192.168.1.1

Testar um domínio:

ping google.com

Limitar a quantidade:

ping -c 4 8.8.8.8

UNIDADE 13 — DNS

13.1 getent hosts

Consulta resolução de nomes:

getent hosts google.com

13.2 dig

Função

Realiza consultas DNS detalhadas.

Instalação:

apt install dnsutils

Consulta:

dig google.com

Consultar somente o endereço:

dig +short google.com

Consultar servidor DNS específico:

dig @8.8.8.8 google.com

13.3 nslookup

Também pode ser utilizado para consultas DNS:

nslookup google.com

UNIDADE 14 — TESTES DE CONECTIVIDADE E ROTAS

14.1 traceroute

Função

Mostra os saltos realizados até um destino.

Instalação:

apt install traceroute

Uso:

traceroute google.com

14.2 tracepath

Alternativa disponível em muitos sistemas:

tracepath google.com

14.3 mtr

Função

Combina características de `ping` e `traceroute`.

Instalação:

apt install mtr-tiny

Uso:

mtr google.com

É muito útil para identificar:

- perda de pacotes;
- latência;
- problemas em determinados saltos.

UNIDADE 15 — PORTAS E SERVIÇOS DE REDE

15.1 ss

Função

Mostra conexões e portas de rede.

Listar portas TCP em escuta:

ss -lnt

Mostrar processos associados:

ss -lntp

Mostrar conexões estabelecidas:

ss -nt

Mostrar TCP e UDP:

ss -tuln

15.2 nc — netcat

Função

Ferramenta extremamente útil para testes de conectividade.

Testar uma porta:

nc -vz 192.168.1.10 22

Testar HTTP:

nc -vz servidor 80

Aplicação

Se o `ping` funciona, mas uma aplicação não conecta, o `nc` pode ajudar a verificar se a porta está acessível.

UNIDADE 16 — SSH

SSH é um dos principais protocolos utilizados na administração remota de servidores Linux.

16.1 ssh

Conectar a um servidor:

ssh aluno@192.168.1.10

Utilizar uma porta diferente:

ssh -p 2222 aluno@192.168.1.10

16.2 scp

Função

Copia arquivos entre computadores utilizando SSH.

Enviar arquivo:

scp arquivo.txt aluno@192.168.1.10:/home/aluno/

Copiar do servidor:

scp aluno@192.168.1.10:/home/aluno/arquivo.txt .

16.3 ssh-keygen

Cria uma chave SSH:

ssh-keygen

A autenticação por chave é muito utilizada em servidores.

16.4 ssh-copy-id

Copia a chave pública para o servidor:

ssh-copy-id aluno@192.168.1.10

Depois disso, dependendo da configuração do servidor, o acesso poderá ser feito sem digitar a senha da conta.

UNIDADE 17 — ARMAZENAMENTO

17.1 df

Função

Mostra espaço disponível nos sistemas de arquivos.

df -h

Exemplo:

```
Filesystem      Size  Used Avail Use%
/dev/sda1        20G   18G  941M  95%
```

Esse é um dos primeiros comandos que um técnico deve utilizar quando o servidor apresenta problema de armazenamento.

17.2 du

Função

Mostra quanto espaço arquivos e diretórios estão utilizando.

du -sh /var

Ver tamanho dos diretórios:

du -sh /var/*

Ordenar:

du -sh /var/* | sort -h

17.3 lsblk

Função

Lista discos e partições.

lsblk

17.4 free

Função

Mostra utilização da memória RAM.

free -h

UNIDADE 18 — DISCOS E SISTEMAS DE ARQUIVOS

18.1 mount

Mostra sistemas montados:

mount

Montar uma partição:

mount /dev/sdb1 /mnt/dados

18.2 umount

Desmontar:

umount /mnt/dados

18.3 lsblk -f

Mostra informações de sistemas de arquivos:

lsblk -f

18.4 blkid

Mostra UUID e tipo de sistema de arquivos:

blkid

UNIDADE 19 — ARQUIVOS COMPACTADOS

19.1 tar

É um dos comandos mais utilizados para criar arquivos de backup.

Criar arquivo `.tar`:

tar -cvf backup.tar /home/aluno

Criar `.tar.gz`:

tar -czvf backup.tar.gz /home/aluno

Extrair:

tar -xzvf backup.tar.gz

Listar conteúdo:

tar -tzvf backup.tar.gz

19.2 gzip

Compactar:

gzip arquivo.txt

Descompactar:

gunzip arquivo.txt.gz

19.3 zip e unzip

Instalar:

apt install zip unzip

Compactar:

zip backup.zip arquivo.txt

Extrair:

unzip backup.zip

UNIDADE 20 — REDIRECIONAMENTO E PIPE

Esses recursos são fundamentais para utilizar o Linux profissionalmente.

20.1 >

Redireciona a saída para um arquivo.

ls > lista.txt

20.2 >>

Adiciona conteúdo ao final.

echo "Novo registro" >> arquivo.txt

20.3 |

O pipe envia a saída de um comando para outro.

Exemplo:

ls -lah | less

Outro exemplo:

ps aux | grep nginx

20.4 tee

Exibe e salva simultaneamente:

ls -lah | tee lista.txt

UNIDADE 21 — grep

Função

Localiza textos dentro de arquivos ou saídas de comandos.

Procurar "erro":

grep "erro" arquivo.log

Ignorar maiúsculas/minúsculas:

grep -i "erro" arquivo.log

Mostrar número da linha:

grep -n "erro" arquivo.log

Pesquisar recursivamente:

grep -r "192.168.1.10" /etc

UNIDADE 22 — SED E AWK

Esses comandos são muito úteis para técnicos que precisam manipular arquivos de configuração e relatórios.

22.1 sed

Substituir texto:

sed 's/antigo/novo/g' arquivo.txt

22.2 awk

Exibir uma determinada coluna:

awk '{print $1}' arquivo.txt

Exemplo:

ip addr | grep inet | awk '{print $2}'

UNIDADE 23 — INFORMAÇÕES DO SISTEMA

23.1 hostname

Mostrar nome da máquina:

hostname

23.2 hostnamectl

Mostrar informações do sistema:

hostnamectl

23.3 uname

Mostrar informações do kernel:

uname -a

23.4 uptime

Mostra há quanto tempo o sistema está ligado:

uptime

23.5 date

Mostra data e hora:

date

UNIDADE 24 — VARIÁVEIS DE AMBIENTE

env

Mostra variáveis de ambiente:

env

---

echo

Exibe valores:

echo $HOME

Exemplo:

echo $PATH

UNIDADE 25 — HISTÓRICO DE COMANDOS

history

Mostra comandos executados:

history

Pesquisar:

history | grep ssh

Executar um comando anterior:

!100

onde `100` é o número exibido pelo `history`.

UNIDADE 26 — FIREWALL

Em servidores Debian modernos, é comum trabalhar diretamente com `nftables` ou utilizar ferramentas que simplificam sua configuração.

26.1 nft

Ver regras:

nft list ruleset

O `nftables` é uma ferramenta poderosa e deve ser configurado com cuidado.

26.2 ufw

Quando instalado e utilizado como camada simplificada:

apt install ufw

Ver status:

ufw status

Permitir SSH:

ufw allow 22/tcp

Permitir HTTP:

ufw allow 80/tcp

Permitir HTTPS:

ufw allow 443/tcp

Ativar:

ufw enable

⚠️ Atenção

Em um servidor remoto, nunca ative ou altere regras de firewall sem garantir que o acesso administrativo continuará funcionando.

UNIDADE 27 — SERVIDORES WEB

Verificar Nginx

systemctl status nginx

Ver portas:

ss -lntp | grep nginx

Testar localmente:

curl http://localhost

Verificar Apache

systemctl status apache2

Testar:

curl http://localhost

UNIDADE 28 — CURL

Função

Realiza requisições para servidores e é muito utilizado para testar APIs e serviços HTTP.

Testar uma página:

curl http://localhost

Mostrar apenas cabeçalhos:

curl -I http://localhost

Seguir redirecionamentos:

curl -L https://exemplo.com

Testar uma API:

curl http://localhost:5000/api

UNIDADE 29 — PROCESSADOR E MEMÓRIA

top

top

htop

htop

free

free -h

uptime

uptime

Uma sequência básica para investigar lentidão:

uptime
free -h
df -h
top

UNIDADE 30 — COMANDOS PARA DIAGNÓSTICO DE PROBLEMAS DE REDE

Quando um servidor não consegue acessar a internet, o técnico pode seguir uma sequência.

1. Verificar interface

ip addr

2. Verificar rota

ip route

3. Testar gateway

ping -c 4 192.168.1.1

4. Testar IP externo

ping -c 4 8.8.8.8

5. Testar DNS

ping -c 4 google.com

6. Consultar DNS

dig google.com

7. Verificar rota

traceroute google.com

8. Verificar portas

ss -tuln

Essa sequência ajuda a separar problemas de:

Interface
↓
Gateway
↓
Internet
↓
DNS
↓
Serviço
↓
Porta

UNIDADE 31 — DIAGNÓSTICO DE ARMAZENAMENTO

Quando o servidor informa que o disco está cheio:

1. Verificar espaço

df -h

2. Verificar inodes

df -i

3. Descobrir diretórios grandes

du -sh /* 2>/dev/null | sort -h

4. Investigar `/var`

du -sh /var/* 2>/dev/null | sort -h

5. Procurar arquivos grandes

find / -type f -size +500M 2>/dev/null

UNIDADE 32 — DIAGNÓSTICO DE SERVIÇOS

Quando um serviço não funciona:

1. Verificar status

systemctl status nginx

2. Verificar logs

journalctl -u nginx

3. Verificar portas

ss -lntp

4. Testar serviço localmente

curl http://localhost

5. Verificar processos

ps aux | grep nginx

UNIDADE 33 — COMANDOS QUE TODO TÉCNICO DEVE DOMINAR

Uma lista prática para memorizar:

pwd
ls
cd
mkdir
touch
cp
mv
rm
cat
less
head
tail
find
grep
chmod
chown
whoami
id
adduser
userdel
passwd
groupadd
usermod
ps
top
htop
kill
systemctl
journalctl
apt
ip
ping
ip route
ss
dig
traceroute
mtr

nc
ssh
scp
df
du
free
lsblk
mount
umount
tar
curl
history

UNIDADE 34 — KIT DE DIAGNÓSTICO RÁPIDO

Em uma situação real de suporte, os seguintes comandos formam um excelente ponto de partida.

Identificação

hostname
whoami
uname -a
uptime

Rede

ip addr
ip route
ping -c 4 8.8.8.8
dig google.com
ss -tuln

Recursos

free -h
df -h
df -i
top

Processos

ps aux

Serviços

systemctl status nome_do_servico

Logs

journalctl -u nome_do_servico

Arquivos

ls -lah
du -sh *
find / -type f -size +500M 2>/dev/null

UNIDADE 35 — COMBINANDO COMANDOS

O verdadeiro poder do Linux aparece quando os comandos são combinados.

Exemplo 1 — Encontrar processos do Nginx

ps aux | grep nginx

Exemplo 2 — Descobrir IP do servidor

ip addr | grep inet

Exemplo 3 — Encontrar arquivos grandes

find / -type f -size +500M 2>/dev/null

Exemplo 4 — Descobrir diretórios que ocupam mais espaço

du -sh /var/* 2>/dev/null | sort -h

Exemplo 5 — Ver portas abertas

ss -tuln

Exemplo 6 — Verificar se o serviço está rodando

systemctl is-active nginx

UNIDADE 36 — ROTINA DE UM TÉCNICO DE INFRAESTRUTURA

Imagine que um usuário informa:

> "O sistema está muito lento."

O técnico não deve simplesmente reiniciar o servidor.

Ele deve investigar.

Etapa 1 — Carga do sistema

uptime

Etapa 2 — Processador

top

Etapa 3 — Memória

free -h

Etapa 4 — Disco

df -h

Etapa 5 — Processos

ps aux

Etapa 6 — Serviços

systemctl --failed

Etapa 7 — Logs

journalctl -p err

O objetivo é **descobrir a causa**, e não simplesmente executar comandos aleatoriamente.

UNIDADE 37 — BOAS PRÁTICAS

Antes de executar um comando

Pergunte:

1. O que esse comando faz?
2. Ele altera o sistema?
3. Ele apaga alguma coisa?
4. Precisa ser executado como `root`?
5. Estou no servidor correto?
6. O caminho informado está correto?
7. Existe backup?

UNIDADE 38 — COMANDOS DE RISCO

Alguns comandos exigem atenção especial.

Exemplos:

rm -rf
chmod -R
chown -R
userdel -r

systemctl stop
systemctl disable
nft
ufw

Esses comandos podem causar perda de dados, indisponibilidade de serviços ou perda de acesso ao servidor.

Um técnico profissional deve sempre verificar o alvo antes de executar.

LABORATÓRIO 1 — PRIMEIROS COMANDOS

Execute:

pwd
ls
ls -la
whoami
hostname
uname -a

Perguntas

1. Qual é seu diretório atual?
2. Qual é o nome do computador?
3. Qual usuário está conectado?
4. Qual versão do kernel está sendo utilizada?

LABORATÓRIO 2 — ARQUIVOS E DIRETÓRIOS

Crie:

laboratorio/
├── documentos/
├── imagens/
├── backups/
└── scripts/

Utilize:

mkdir -p laboratorio/documentos
mkdir -p laboratorio/imagens
mkdir -p laboratorio/backups
mkdir -p laboratorio/scripts

Depois:

touch laboratorio/documentos/anotacoes.txt
touch laboratorio/scripts/teste.sh

Verifique:

tree laboratorio

LABORATÓRIO 3 — USUÁRIOS E GRUPOS

Crie um grupo:

sudo groupadd laboratorio

Crie três usuários:

sudo adduser aluno01
sudo adduser aluno02
sudo adduser aluno03

Adicione os usuários ao grupo:

sudo usermod -aG laboratorio aluno01
sudo usermod -aG laboratorio aluno02
sudo usermod -aG laboratorio aluno03

Verifique:

groups aluno01

LABORATÓRIO 4 — PERMISSÕES

Crie:

mkdir projeto
touch projeto/arquivo.txt

Verifique:

ls -l projeto

Altere as permissões:

chmod 755 projeto

Depois:

chmod 644 projeto/arquivo.txt

Observe a diferença.

LABORATÓRIO 5 — DIAGNÓSTICO DE REDE

Execute:

ip addr

Depois:

ip route

Teste o gateway.

Depois:

ping -c 4 8.8.8.8

E:

ping -c 4 google.com

Compare os resultados.

Desafio

Se `8.8.8.8` responder, mas `google.com` não responder, qual componente provavelmente deve ser investigado?

LABORATÓRIO 6 — DIAGNÓSTICO DE SERVIÇO

Escolha um serviço instalado no laboratório.

Exemplo:

systemctl status ssh

Depois:

ss -lntp

E:

journalctl -u ssh

Identifique:

- se o serviço está ativo;
- qual porta utiliza;
- quais mensagens aparecem nos logs.

LABORATÓRIO 7 — SERVIDOR COM DISCO CHEIO

Imagine que o monitoramento informe:

DISK CRITICAL
Free space: 941 MiB

Investigue utilizando:

df -h

Depois:

df -i

Depois:

du -sh /* 2>/dev/null | sort -h

Depois:

du -sh /var/* 2>/dev/null | sort -h

Desafio

Descubra qual diretório está ocupando mais espaço e investigue seu conteúdo.

LABORATÓRIO 8 — SERVIDOR WEB

Instale o Nginx:

sudo apt update
sudo apt install nginx

Verifique:

systemctl status nginx

Verifique a porta:

ss -lntp | grep :80

Teste:

curl http://localhost

LABORATÓRIO 9 — SSH

Descubra o endereço IP:

ip addr

Em outro computador, tente:

ssh usuario@IP_DO_SERVIDOR

Depois copie um arquivo:

scp arquivo.txt usuario@IP_DO_SERVIDOR:/home/usuario/

DESAFIO FINAL — TÉCNICO DE INFRAESTRUTURA

Imagine o seguinte chamado:

> **Servidor WEB01 está apresentando lentidão e alguns usuários não conseguem acessar o sistema.**

Você deverá investigar sem reiniciar o servidor inicialmente.

Utilize:

hostname
uptime
free -h
df -h
ip addr
ip route
ping -c 4 8.8.8.8
ss -lntp
systemctl status nginx
journalctl -u nginx
top

O aluno deverá descobrir:

- nome do servidor;
- tempo ligado;
- utilização de memória;
- utilização do disco;
- endereço IP;
- gateway;
- conectividade;
- portas em utilização;
- estado do servidor web;
- mensagens de erro;
- processos que consomem recursos.

PROJETO INTEGRADOR

Montando um pequeno servidor Linux

O aluno deverá montar um servidor Debian capaz de:

1. possuir usuários diferentes;
2. possuir grupos;
3. utilizar permissões;
4. disponibilizar SSH;
5. executar um servidor web;
6. possuir endereço IP configurado;

7. permitir testes de conectividade;
8. possuir logs;
9. possuir ferramentas de monitoramento;
10. realizar backup de arquivos.

Comandos que deverão ser utilizados

adduser
groupadd
usermod
chmod
chown
apt
systemctl
journalctl
ip
ping
ss
ssh
scp
df
du
free
tar
curl

TABELA DE CONSULTA RÁPIDA

Categoria	Comando	Função
Navegação	`pwd`	Mostra diretório atual
Navegação	`ls`	Lista arquivos
Navegação	`cd`	Muda diretório
Arquivos	`mkdir`	Cria diretório
Arquivos	`touch`	Cria arquivo
Arquivos	`cp`	Copia
Arquivos	`mv`	Move/renomeia
Arquivos	`rm`	Remove
Texto	`cat`	Exibe arquivo
Texto	`less`	Visualiza arquivo
Texto	`head`	Início do arquivo

Texto	`tail`	Final do arquivo
Busca	`find`	Localiza arquivos
Busca	`grep`	Localiza texto
Permissões	`chmod`	Altera permissões
Permissões	`chown`	Altera proprietário
Usuários	`adduser`	Cria usuário
Usuários	`userdel`	Remove usuário
Usuários	`passwd`	Altera senha
Grupos	`groupadd`	Cria grupo
Grupos	`usermod`	Modifica usuário
Processos	`ps`	Lista processos
Processos	`top`	Monitora processos
Processos	`htop`	Monitora processos
Serviços	`systemctl`	Administra serviços
Logs	`journalctl`	Consulta logs
Pacotes	`apt`	Gerencia pacotes
Rede	`ip`	Configuração/informações de rede
Rede	`ping`	Testa conectividade
Rede	`ss`	Verifica conexões/portas
DNS	`dig`	Consulta DNS
Rede	`traceroute`	Mostra rota
Rede	`mtr`	Analisa rota/latência
Rede	`nc`	Testa portas
Remoto	`ssh`	Acesso remoto
Remoto	`scp`	Copia arquivos via SSH
Disco	`df`	Espaço disponível

Disco	`du`	Espaço utilizado
Disco	`lsblk`	Lista discos
Memória	`free`	Memória RAM
Backup	`tar`	Compactação/arquivamento
Web	`curl`	Testa serviços HTTP
Sistema	`hostname`	Nome da máquina
Sistema	`uname`	Informações do kernel
Sistema	`uptime`	Tempo ligado
Histórico	`history`	Histórico de comandos

CONCLUSÃO

O objetivo desta apostila não é fazer o aluno decorar centenas de comandos.

O objetivo é desenvolver uma forma de pensar como um técnico de infraestrutura.

Quando um problema aparecer, o aluno deverá aprender a perguntar:

O sistema está funcionando?
↓
O serviço está funcionando?
↓
O processo está funcionando?
↓
A porta está aberta?
↓
A interface está funcionando?
↓
Existe conectividade?
↓
O DNS está funcionando?
↓
Existe espaço em disco?
↓
Existe memória disponível?
↓
O que dizem os logs?

Linux é muito mais do que comandos.

É uma ferramenta para **investigar, diagnosticar, administrar, automatizar e solucionar problemas de infraestrutura**.